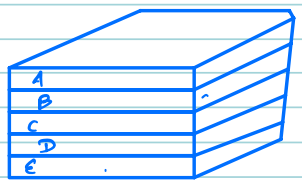
Memory Management

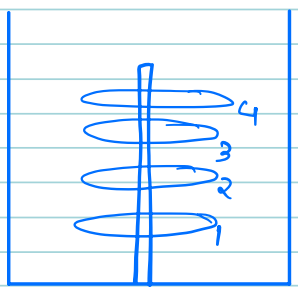
* Introduction about Stack
  → Basic Meaning

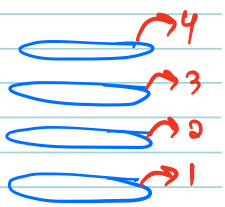
Some Examples

① Stack of Books



② Idli Cooker





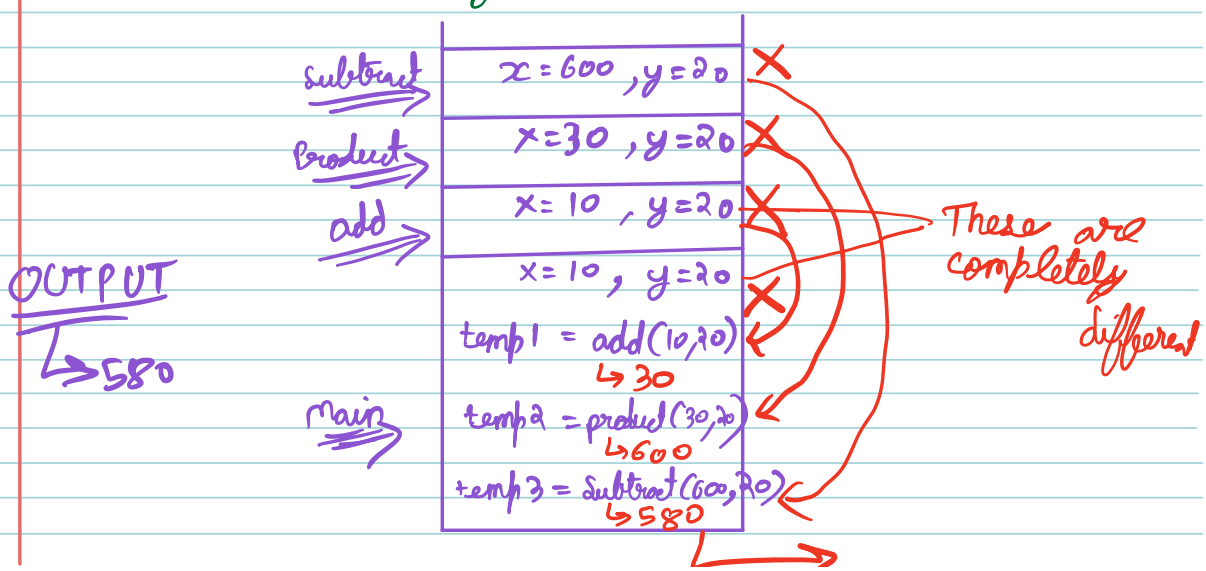
* LIFO → Last in first out
* Insert & Delete happens from top.
* you always access of topmost element.

Stack :- container which follows above properties & hold data

Question :- what will be the output of this code.

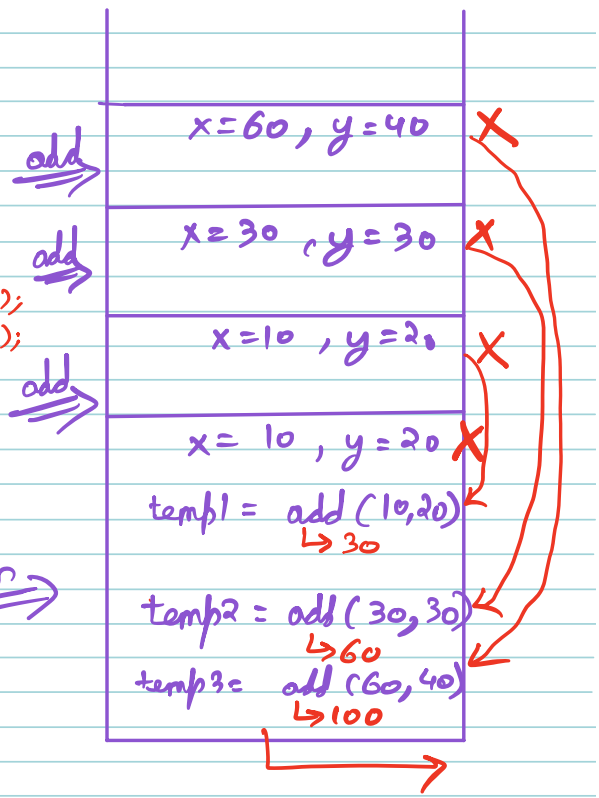
```
int add ( int x , int y ) {
        return x + y ;
}

int product ( int x , int y ) {
        return x * y ;
}

int subtract ( int x , int y ) {
        return x - y ;
}

Public static void main ( ) {
        int x = 10;
        int y = 20;
        int temp1 = add (x, y) ;
        int temp2 = product ( temp1 , y );
        int temp3 = subtract ( temp2 , y );
        sout( temp3);
}
```

★ Simple Dry Run ⇒ output : 580



Question :- Consider the below code:

```
int add ( int x , int y){
    return x + y;
}

Public static void main(){
    int x = 10
    int y = 20
    int temp1 = add (x, y);
    int temp2 = add ( temp1 , 30);
    int temp 3 = add ( temp 2, 40);
    Sout ( temp3) ;
}
```

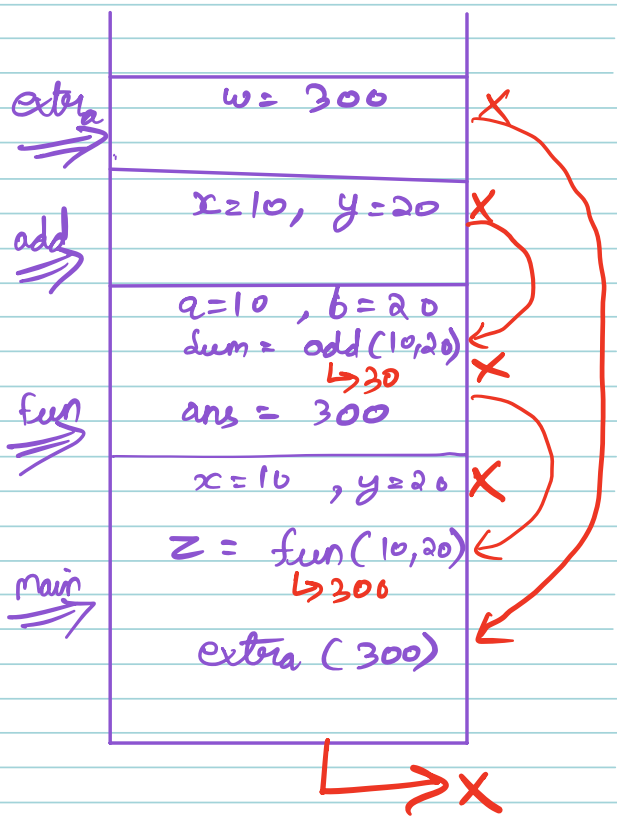


OUTPUT

100

Question :- Consider the below code :-

```
int add (int x, int y){
    return x+y;
}

static int fun (int a, int b){
    int sum = add (a,b);
    int ans = sum * 10;
    return ans;
}

static void extra (int w){
    sout ("Hello");
    sout (w);
}

Public static void main(){
    int x = 10;
    int y = 20;
    int z = fun (x,y);
    sout (z);
    extra (z);
}
```



OUTPUT

300
Hello
300

#Types Of Memories

↳ whenever Java program runs, it takes space in RAM. This space is further divided:-

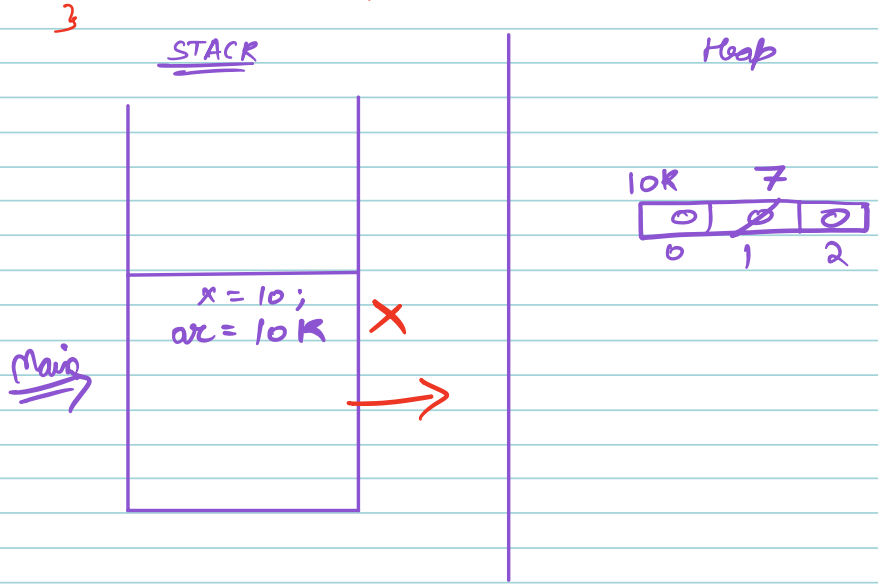
(1) Stack Memory :- All the primitive data type & reference will be stored in stack.

(2) Heap :- Objects / Container of that reference is stored in Heap. Arrays, Arraylist, objects are created inside Heap.

★ Anything defined using "new" keyword creates Object.

Question :- Consider the below code :-

```
Public static void main ()s
{
    int x=10;
    int [] ar = new int [3];
    SOUT ( ar);
    SOUT ( ar [2]);
    arr [1] = 7;
}
```



OUTPUT

10K

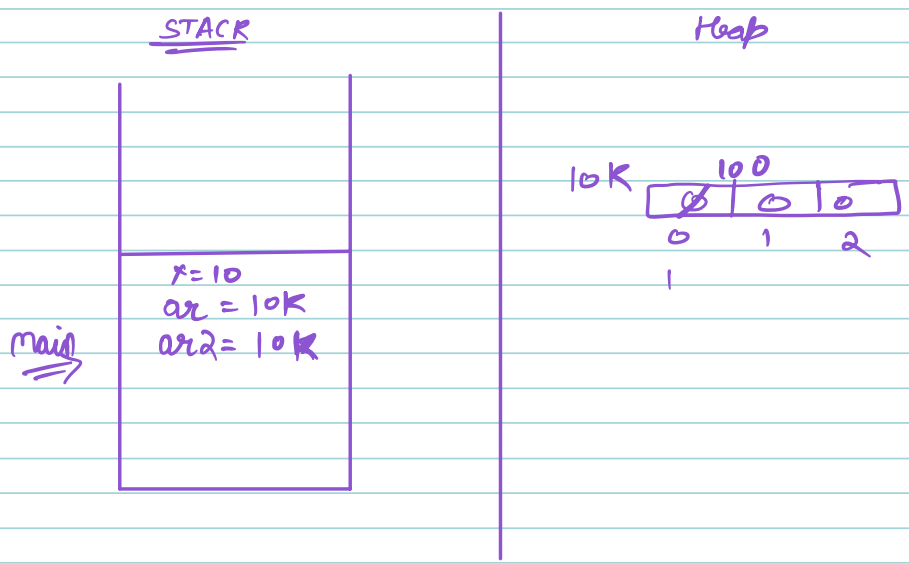
0

↳ Basics about the above Content.

1. Primitives are stored in stack.
2. References / address is stored in stack.
3. objects are stored in heap

Question :- Consider the below code.

```
Public static void main () {
    int x = 10;
    int[] ar = new int [3];
    int [] ar2 = ar;
    ar [0] = 100;
    SOUT (ar);
    SOUT (ar2);
    SOUT (ar [0]);
```

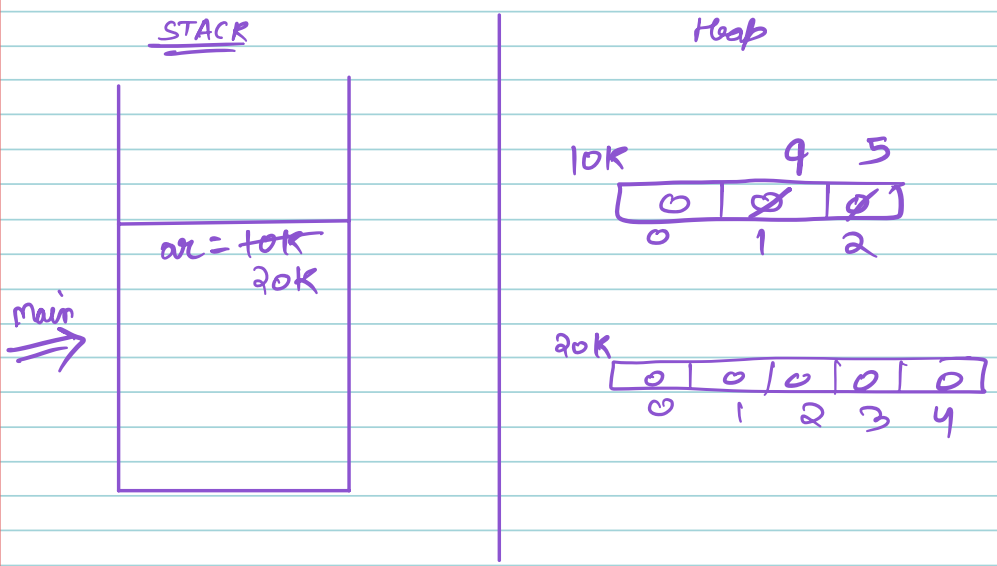


OUTPUT

10K
10K
100

Question Consider the below code

```
Public static void main(){
    int[] ar = new int[3];
    SOUT (ar)
    ar[1] = 4;
    ar[2] = 5;

    ar = new int[5];
    SOUT (ar);
```

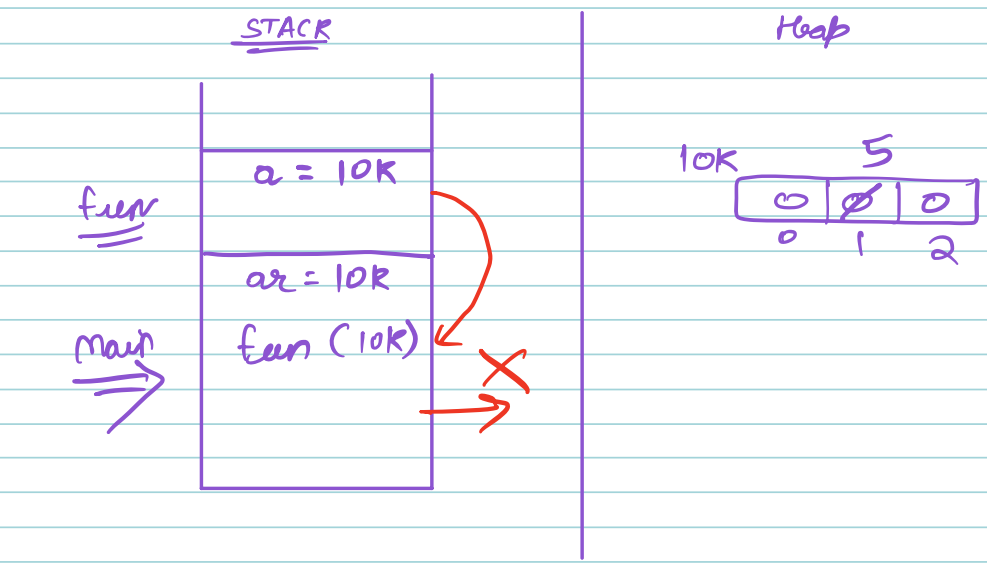


OUTPUT

10K
20K

Question : Consider the code below

```
static void fun(int[] a){
    SOUT(a);
    a[1] = 5;
}

public static void main(){
    int[] ar = new int[3];
    SOUT(ar);
    ar[0];
    ar[1];
    fun(ar);
    SOUT(ar[1]);
}
```

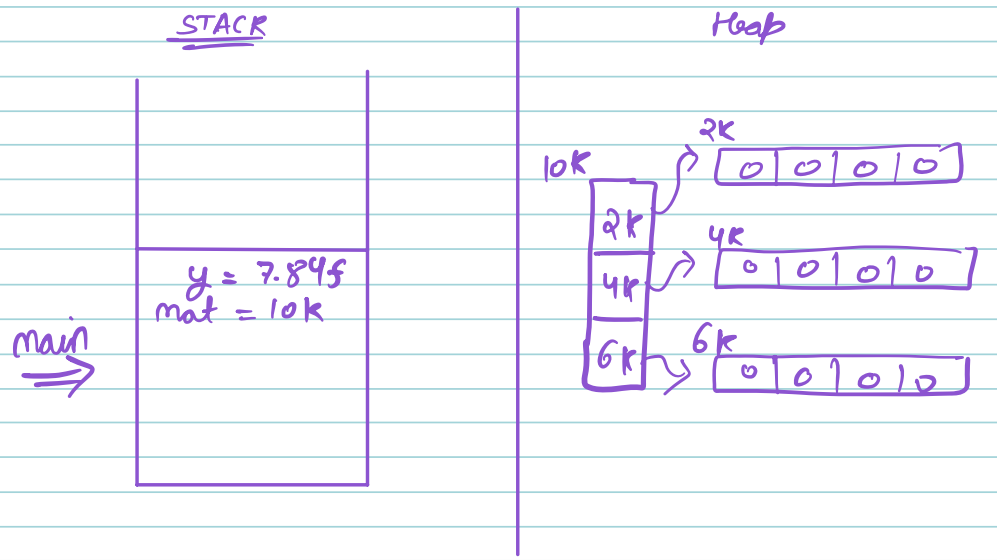


OUTPUT

10K
10K
5

Question :- Consider the code below

```
public static void main() {
    float y = 7.84f;
    int[][] mat = new int[3][4]
    sout (mat);
    sout (mat[1]);
    sout (mat[1][3]);
}
```

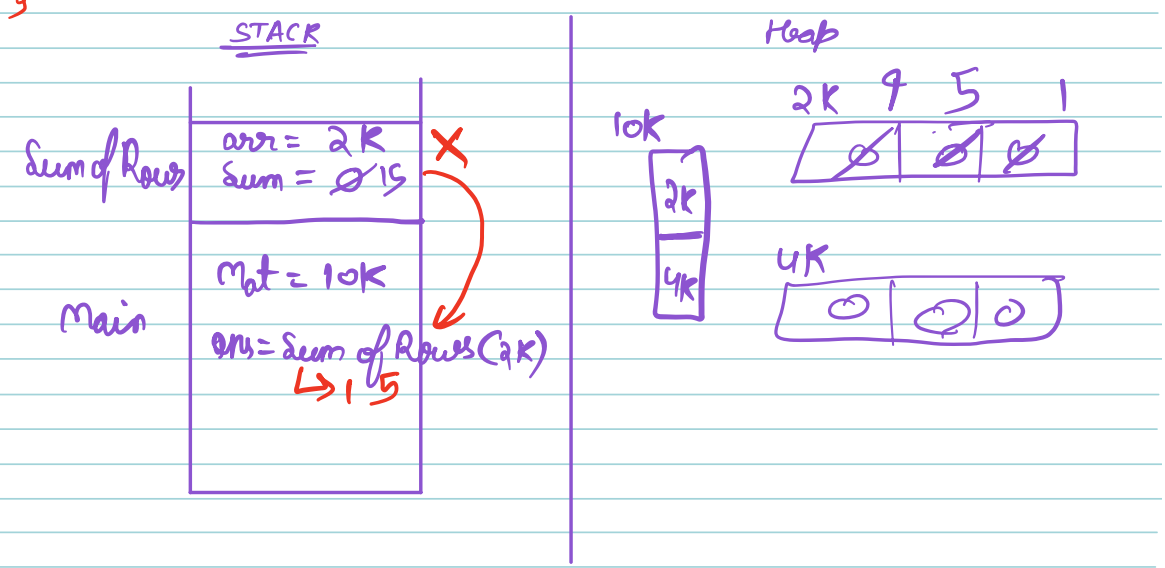


OUTPUT

10 K
4 K
0

Question :- Consider the code below

```
Static int SumOfRow ( int[] arr) {
    SOUT (arr);
    int Sum = 0;
    for ( i = 0; i < arr.length ; i++) {
        Sum = Sum + arr[i];
    }
    return Sum;
}

Public Static Void Main () {
    int[][] mat = new int [2][3];
    mat[0][0] = 9;
    mat[0][1] = 5;
    mat[0][2] = 1;
    int ans = SumOfRow ( mat[0]);
    SOUT (ans);
}
```



OUTPUT ⇒ 2K
15

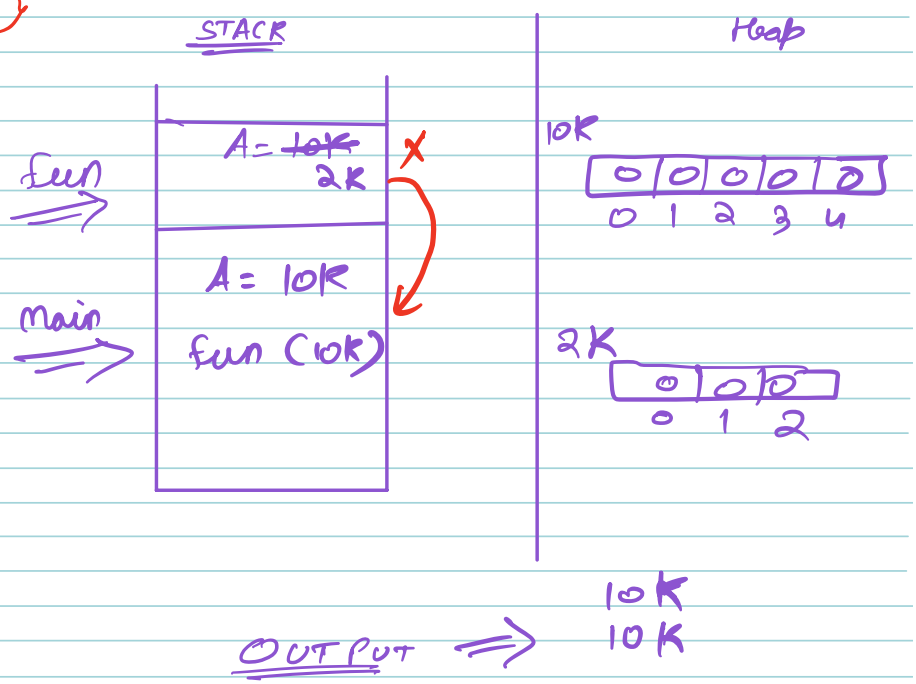
IMP INFO

↳ In Java everything is pass by value.

↳ Pass by value means pass copy of data to function called.

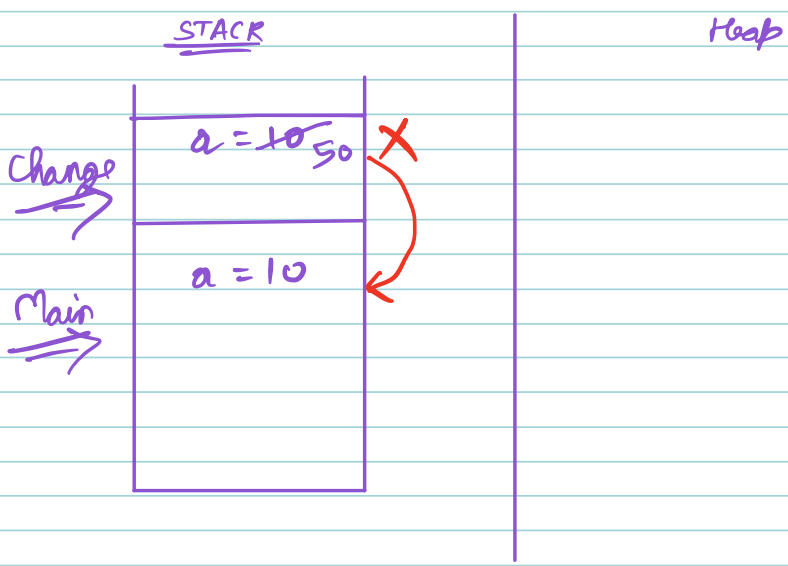
Ex

```
fun ( int[] A) {
    A = new int[3];
}

main ( ) {
    int[] A = new int[5];
    SOUT ( A );
    fun (A) ;
    SOUT (A);
}
```



Quiz! :- Predict the output

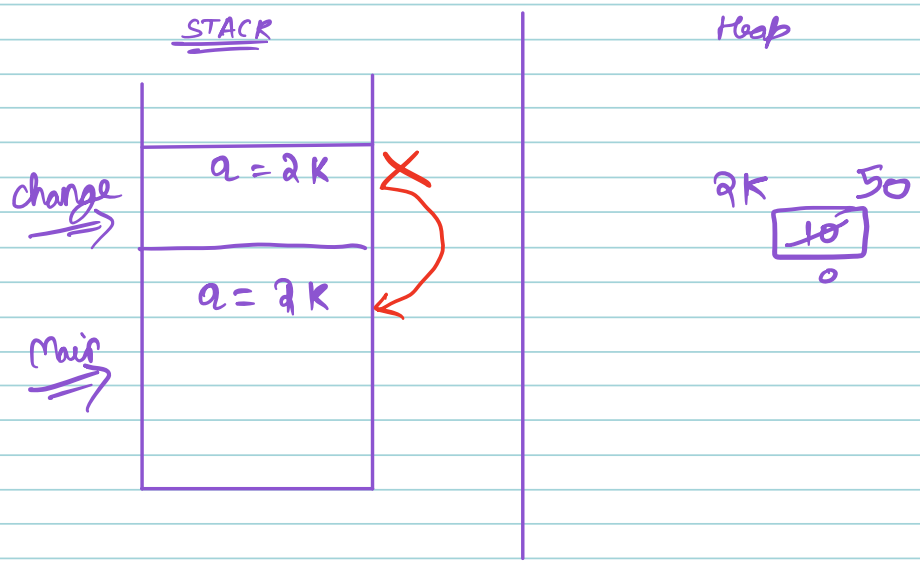
```
static void change(int a){
    a=50;
}

public static void main(){
    int a = 10;
    change(a);
    sout(a);
}
```



OUTPUT :— 10

Quiz2 :- Predict the output

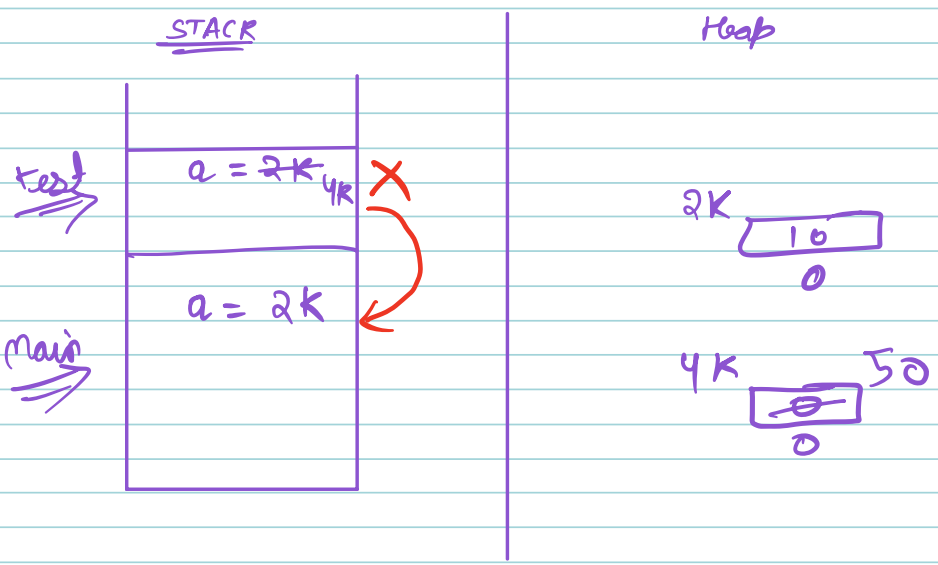
```
void change ( int [] a){
    a[0] = 50;
}

Public static void main() {
    int[] a = {10};
    change (a);
    SOUT ( a[0]);
}
```



OUTPUT :⇒ 50

Quiz 3 : Predict the output

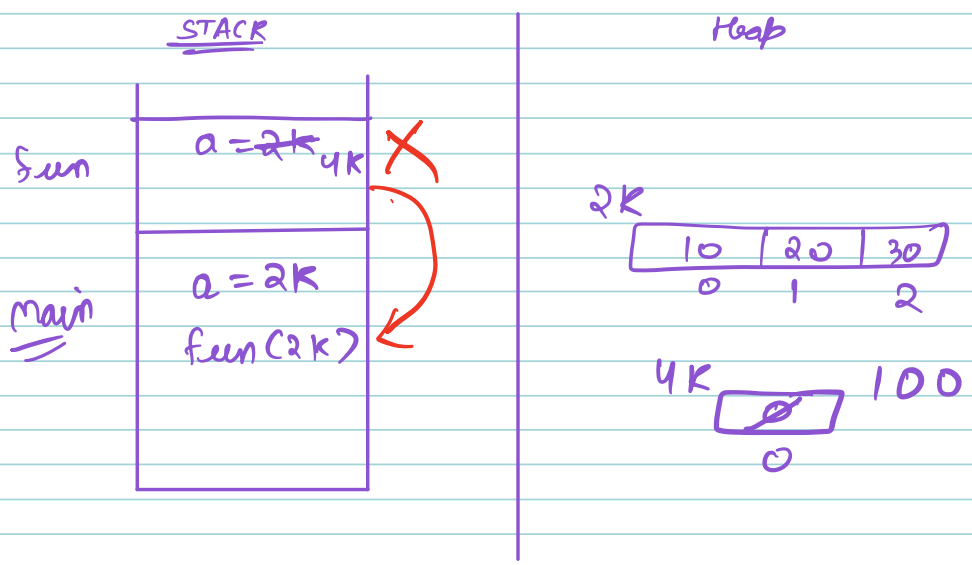
```
void test ( int[] a){
    a = new int[1];
    a[0] = 50;
}

Public Static void Main(){
    int[]a = {10};
    test (a);
    SOUT (a[0]);
}
```



OUTPUT : 10

Quiz 4 Predict the output

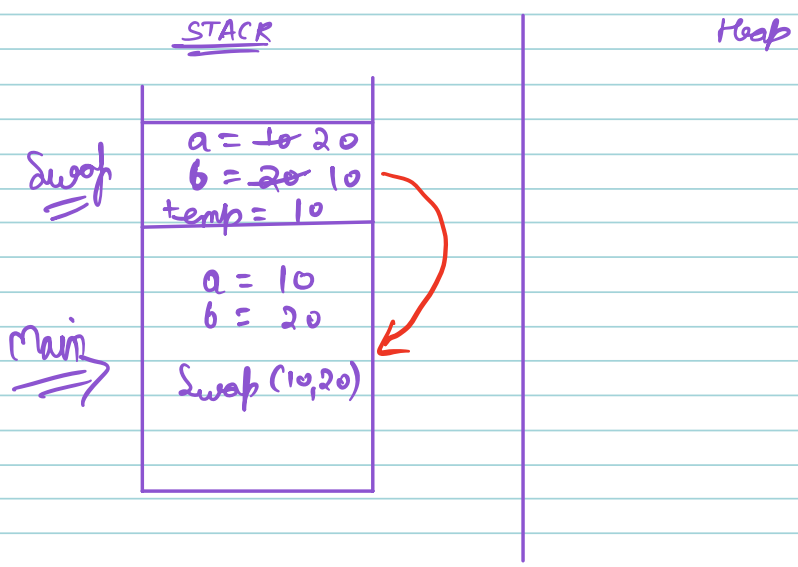
```
void fun ( int[] a) {
    a = new int[1];
    a[0] = 100;
}

public static void main() {
    int[] a = {10, 20, 30};
    fun(a)
    SOUT ( a[0]);
}
```



OUTPUT :- 10

Quiz 5 :- Predict the output

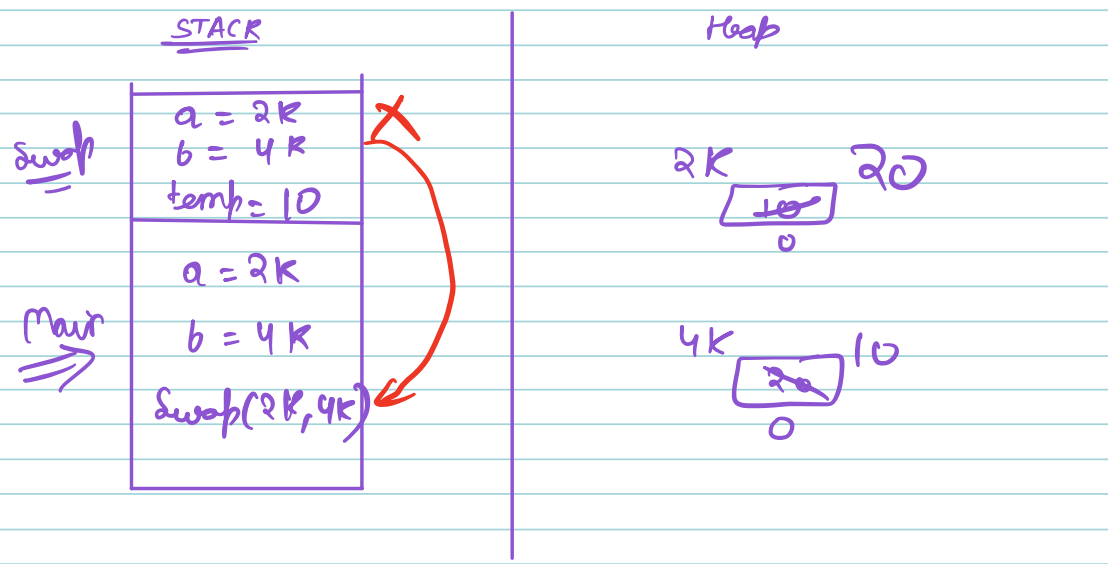
```
void Swap ( int a, int b) {
    int temp = a
    a = b;
    b = temp;
}

public static void main () {
    int a = 10;
    int b = 20;

    swap (a, b);
    SOUT ( a + " " + b);
}
```



OUTPUT : 10 20

Quiz 6 : Predict the output

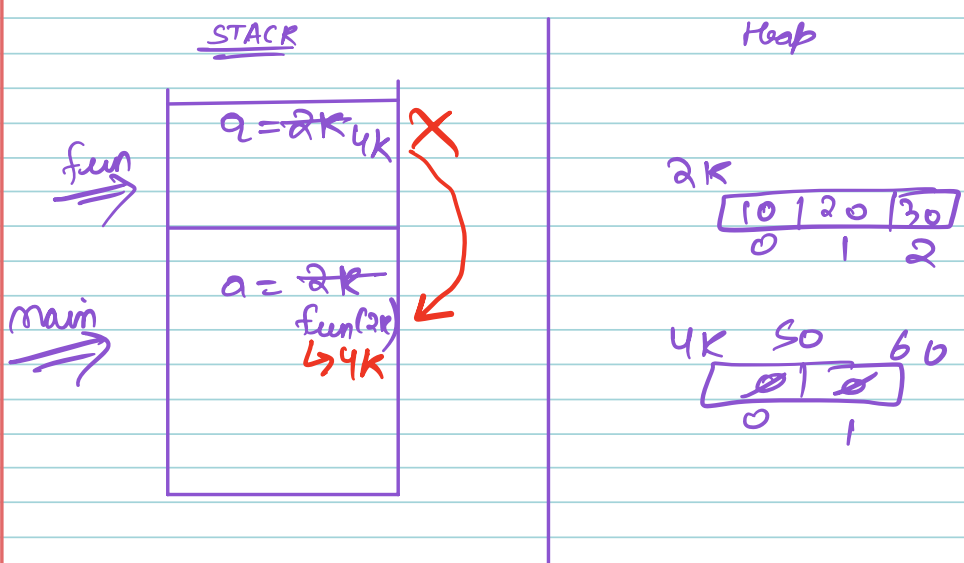
```
void Swap (int[] a, int [] b){
    int temp = a[0];
    a[0] = b[0];
    b[0] = temp;
}

public static void main (){
    int[] a = {10};
    int[] b = {20};
    Swap (a, b);
    SOUT (a + " " + b);
    SOUT (a[0] + " " + b[0]);
}
```



OUTPUT 20 10

Quiz 7 :- Predict the output

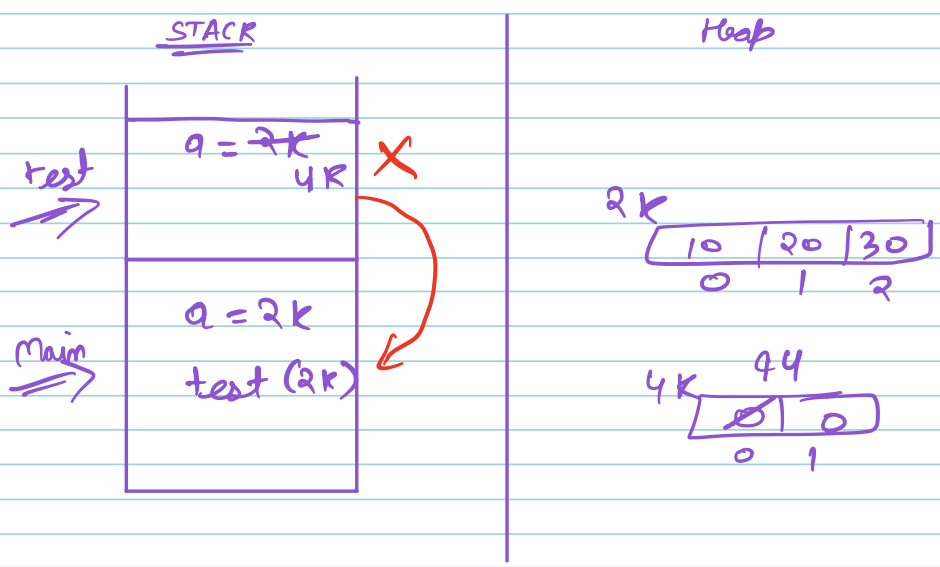
```
int[] fun (int[] a){
    a = new int[2];
    a[0] = 50;
    a[1] = 60;
    return a;
}

public static void main() {
    int[] a = {10, 20, 30};
    a = fun(a)
    SOUT (a[0]);
}
```



OUTPUT : 50

Quiz 8 :- Predict the output

```
void test (int[] a){
    a = new int[2];
    a[0] = 94;
}

Public static void main () {
    int[] a = {10, 20, 30};
    test (a);
    SOUT (a[0]);
}
```



OUTPUT : 10